DOCUMENT RESUME

ED 302 195 IR 013 554

AUTHOR Nicol, Anne; And Others
TITLE Multimedia in Education: Thought Pieces.
PUB DATE Jun 86
NOTE 23p.; Papers presented at an Invitational Conference on Multimedia in Education (Cupertino, CA, June 19-20, 1986). For the complete journal, see IR 013 538.
PUB TYPE Journal Articles (080) -- Speeches/Conference Papers (150)
JOURNAL CIT Learning Tomorrow: Journal of the Apple Education Advisory Council; n3 p324-353 Spr 1987

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Authoring Aids (Programing); Computer Managed Instruction; *Computer System Design; *Design Requirements; Educational Technology; *Learning Processes; *Multimedia Instruction; *Optical Disks; Videodisks
IDENTIFIERS *Hypermedia

ABSTRACT

The six "thought pieces" presented in this document summarize the personal perspectives of conference participants on issues raised in formal or information discussions, but not part of the central themes of the conference or conference papers. They include: (1) "Assuming That..." (Anne Nicol, Apple Computer, Inc.); (2) "A Layered Theory of Design for Optical Disc Software" (Mike Liebhold, Apple Computer, Inc.); (3) "Links or Stories: A Compromise" (Bernard Frischer, University of California at Los Angeles); (4) "Getting a Boost from the Real World" (Ted Kaehler, Apple Computer, Inc.); (5) "Authoring with Hypermedia" (Carol Kaehler, Apple Computer, Inc.); and (6) "Navigation in Large Knowledge Spaces" (Steven Weyer, Apple Computer, Inc.). (EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

X This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Thought Pieces

BEST COPY AVAILABLE

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Saul Rockman

TO THE EDU NAL RESOURCES INFORMATION CENTER (ERIC)."

A number of issues raised at the Conference did not form the central theme for any paper. Many of these were raised in our formal discussion sessions or in hallway discussions. Others were raised by individuals who attended the Conference but did not present papers.

In order not to lose these perspectives, Conference attendees were invited to write "thought pieces," which summarized their personal perspectives on these "extra issues" in a few pages. These are included in this section.

Assuming That...

Anne Nicol
Apple Computer, Inc.

New technologies—in particular, the integrated sorts of systems described in this book—hold great promise for learning. Their linking capabilities invite individually customized and inquiry-driven approaches to knowledge acquisition. Their interactivity encourages problem-solving activities. And their capacity for vast storage and quick retrieval puts whole domains of visual and auditory information within reach. Potentially, they become the tools that enable people to learn in a most meaningful and effective way—by discovery.

Somehow, we all know that discovery learning is "good" learning. Things that we have found out for ourselves take on special importance; the knowledge we gain is bolstered by personal and emotional content. We remember what we have discovered longer, and we are more likely to pay attention to new, related information. But discovery is not always the most effective way to learn a particular concept or to achieve a given learning objective. And even when it is, circumstances in the situation may obstruct the natural learning process. As designers, we need to carefully examine and test the assumptions that underlie both the "what" and the "how" of our projects.

Jerome Bruner, an eloquent advocate of discovery learning, reminds us that "discovery favors the well-prepared mind" (Bruner, 1962). His argument is that there are relevant prerequisite experiences that, in any domain, make discovery more likely. Let us explore briefly what the implications of Bruner's statement might be for our new interactive multimedia projects.

Because the users of these systems will often be learning how to use the system at the same time they are learning the curriculum delivered by the system, there are really two classes of prerequisite experience that need attention. First, navigating through a complex learning environment will require an understanding of how navigational features of the user interface may correspond to the organization of information in the system. For example, the use of arrows to move through levels of information is intuitive only to the learner who already knows that the information might be organized in levels and who knows something meaningful about the relationship between levels. This kind of knowledge need not be specific to the discipline or to the organization of the particular data base. It may come from previous experience in using other information systems, or from a good advance overview or map of the system at hand.

A second class of assumptions has to do with relationships between the learning process and the subject matter. Most programs that encourage an inquiry approach assume that learning begins with a learner's question. But as experienced teachers know, students new to a subject are rarely able to formulate the questions that will lead to fruitful research. Questions of that sort must be based on hypotheses, and it takes some practice and some knowledge of the field to generate useful hypotheses. The system needs to have ways of modeling the inquiry process for the learner in the same way that good instructors lead their students to ask appropriate questions.

Also, in designing responsive systems, we make assumptions about the motivation and interest of learners. Taking full advantage of the opportunities made available by new technologies requires "mindfulness" on the part of the learner. "This mindfulness—the deliberate employment of nonautomatic processes—is partly determined by learners' proclivities, perceptions, expectations, fears, and self-perceptions, and partly by their immediate experiences with the computer" (Salomon, 1985). The latter we can have control over; the former, we must be ever mindful of ourselves.

Once one develops expertise in an area, it is always difficult to remember what one "didn't know then." But we can get lots of help by trying out our systems with "real" students and by working with "real" teachers throughout. We can do the same sorts of task analyses that curriculum writers have always done, in which we identify prerequisite concepts, both cognitive and procedural, for each important learning objective. To conclude, this is a call for continuing, formative evaluation of our design work, in which we test our assumptions about learners as well as our program content against educational goals. In this way, we ensure that the tools we create will truly enhance the learning process for all learners.

References

Bruner, Jerome. (1962) *On Knowing: Essays for the Left Hand.* Cambridge: Harvard University Press.

Salomon, Gavriel. (1985) "Information technologies: What you see is not (always) what you get." Paper presented at the Annual Meeting of the American Educational Research Association.

Anne Nicol is an educational psychologist whose research interests are children and television, children and computers, and human-computer interactions. She has taught kids of all ages how to use computers and is currently directing the Human Interface Testing Laboratory at Apple Computer, Inc.

A Layered Theory of Design for Optical Disc Software

Mike Liebhold
Apple Computer, Inc

8

Introduction

The following paper is a summary of thoughts after the Multimedia in Education conference at Apple Computer in 1986.

Most applications of CD-ROM are transformed network-information-retrieval products. Interactive videodiscs, on the other hand, have been principally designed for structured computer-aided instruction (CAI). Both, however, neglect the promise and ultimate power of the technology.

CD-ROMs that help a user merely find *facts* may not offer added value over print and network services. The media and computational technology are clearly well suited for *decision aid*. New models of productivity imply more intelligence than mere search and retrieval.

Videodiscs that have been hard-wired, tree-structured CAI programs ignore the vast archival capabilities of the format. A videodisc, for example, with 27 minutes of discrete video sequences still has a capacity for 5,400 still frames—a capacity rarely utilized. *Videodisc is a data base technology* that may also be used as an instructional resource.

The Four Layers: An Approach to Optical Disc Software Design

In this paper, I propose a design approach that combines the archival power of optical media with the intelligence and flexibility of computer software. This approach entails four layers:

- Data
- Index
- Browser
- Intelligence

Data Layer

Any combination of five principal data types may be combined on a compact or LV (laservision standard) disc:

Text (usually ASCII)
Graphics (including fax, bit maps, vector drawings, and GKS)
Motion (animation, video {ntsc, pal, secam, hdtv, digital})
Sound (speech, music, sampled sound)
Code (computer programs, machine instructions)

The key words for this layer are *appropriate* and *coherent*. Obviously, if you are planning a disc on medieval history, footage of the Voyager planetary flyby would not be useful.

Index Layer

A *data base* management system of the data on the disc constitutes the index layer. Following Ted Nelson's Xanadu hypertext concepts, the system should permit *annotation*, *updates*, and *personalization*. Imagine a photograph of the earth taken from a satellite in 1978. A modern meteorologist who understands a previously unknown cloud formation because of recent research would clearly want to update the data-base record for the optical disc image.

A personally annotated version of the library card catolog is also possible. Since we are considering read-only media, the *differential files* will have to reside on a magnetic-media floppy or hard disk. You wouldn't be carrying the entire card catalog around on your floppy, just the *pointers* to the cards you have annotated.

Browser Layer

Any disc-based information system may include a wide range of tools or methods such as the following for finding or navigating through data:

- Key word(boolean) search
- Alphanumeric search
- Topical browsers
- Graphic browsers
- Intelligent browsers

Key word (boolean) search.
An example would be "Find Saturn *and* moons" or "Find Lincoln *not* automobile." This is a technique fairly familiar to librarians and network data-base searchers. The most commonly developed application for CD-ROM, it really needs an intelligent thesaurus and natural-language understanding to be fully effective in other words, supporting the *DWIM* command—*Do what I mean!*)

Alphanumeric search.
As in a book index, you have **abacus** under A, **baseball** under B, and **captain** under C. You need to know the name of what you're looking for in order to find it.

Topical browsers.
A table of contents is a topical browser. Vast files of data on a CD or videodisc may be comfortably navigated by chapter, unit, and section or by various subject or theme topics. Using various pull-down and pop-up menu schemes on Xerox and Symbolics Lisp machines, Steve Weyer has demonstrated great examples of sophisticated, structured topical-browsing systems.

Graphic browsers.
Maps, diagrams, and charts are all potentially useful as an interface to data. Point and click on Chicago, Illinois on the map, and the appropriate data-base record or picture pops up. Filevision is an early example of this technique. The MIT movie manual projects produced by Steve Gano, Andy Lippman and others illustrate the rich variety of applications of digitized images for browsing and instruction.

Transparent graphic-overlay in an interactive video environment may, in the future, permit running video objects to be directly manipulated. As a video object moves across the screen, a transparent "cursor target" follows: hypermedia! New 3-D graphics systems also suggest a new opportunity fo direct manipulation of screen "objects."

Intelligent browsers.
Real-world knowledge is not linear. Three-dimensional arrays or *concept networks* of common-sense links are closer to the way humans organize thoughts. Ted Kaehler calls them *semantic networks*. Here's Ted's example:

> A *fern* in the forest >
>A *tree* in the forest >
> A *logger* cuts the tree >
> The tree is taken on a truck to the *sawmill* >
> The tree is cut into *lumber* >
> The lumber is used to construct a *building* >
> The building was designed by an *architect*, >
> The architect has a computer >the computer has a CRT > > > etc.

We've traveled from a fern to a CRT in comfortable, lateral steps.

Although some work is being done at MCC and elsewhere to automate construction, accumulation of these links will most likely be an incremental manual and social process for some time. The emergence of common hypertext environments will permit communities of scholars to aggregate their knowledge into machine-readable formats. Greg Crane's Perseus project at Harvard is a fine example of scholars (in this case, Greek history scholars) collaborating to construct a hypertext and hypermedia representation of a discipline.

The Intelligent Layer

Any of the following types of applications may be layered on top of an optical-disc-based reference system:

- Instruction
- Simulation
- Productivity
- Creativity
- Serendipity
- "Expert"

While most videodisc producers have ignored potential reference value of the browser layer and gone straight for interactive instruction, most CD-ROM creators have stopped at the browser layer—short of productive applications other than information retrieval.

Clearly, the future of optical disc information systems rests in the development of new *data-rich* models of productivity. A *hyper-spreadsheet* model of world financial phenomena, if well designed, would be far more useful then a CD-ROM almanac of financial facts, and an encyclopedia with curriculum-coordinated courseware more useful than a basic reference. A stand-alone page-layout program would be considerably enhanced by a library of images, and an *intelligent data-base agent* would be more helpful than a vanilla terminal program, and a *simulation* of a living biosystem more compelling than a collection of photos and files.

Mike Liebhold is Manager of Optical Media Applications in the Advanced Development Group of Apple Computer Inc. He is developing hardware and software systems for optical media-based information products. Previously he developed publishing relationships for optical media products with publishers, database companies, and optical software developers. Before joining Apple Computer he was responsible for advanced market development at ByVideo Inc., a transactional videodisc system vendor. At the Atari Division of Warner Communications, he was responsible for the Strategic Systems Research Laboratory and its work on optical information systems.

Links or Stories— A Compromise

Bernard Frischer
UCLA

14

All designers of multimedia computer systems for education have to confront a basic question at some point in their work. Should end users be able to range freely through the various data bases, like scholars in a library, making whatever connections, or "links," they wish between the data, or should they be led along a predetermined path through the material to be mastered? This question is, of course, but a specific form of the fundamental question all teachers must answer—no matter what subject they are teaching or what technology they are using—about whether a student learns better in a free or a structured environment.

The advantages and disadvantages of freedom versus structure are well-known. Freedom enables students to find themselves, and nothing is more important to individual development than that; structure can be counterproductive because it forces students to learn what they may not yet be interested in learning. Structure ensures that students learn *something*, however unwillingly; freedom entails the possibility that students will never get off the mark and find either themselves or their way through the desired material. And so on.

It seems clear that the major advantage of computers is precisely their programmability, or their flexibility to interact with users in ways that traditional tools cannot. A riveting hammer cannot become a machinist's hammer—let alone a basin wrench—just because the job demands a different tool. Obvious as this observation is, it has, I daresay, been frequently ignored in computer applications to education, perhaps because of the genesis of CAI in theories of managed instruction.

In computer-managed instruction, "individualization" is defined simply as giving students the power to make their way *at their own speed* along a predetermined route.[1] Individualization might, however, also be defined as determining how a student best learns before deciding whether to impose structure or permit freedom. Assuming that a choice must be made between self-directed and structured learning on the computer, then good instructional design ought to begin with a profiling of the student to see which approach is appropriate *for that student at that time for that subject*. The absence of such profiling is a major factor, I think, contributing to the otherwise puzzling failure of most CAI to show statistically significant results. By imposing "one right way" on all students, most CAI measures up as a wash because it turns students on or off in about the same percentage of cases.

[1] See F.B. Baker, *Computer Managed Instruction Theory and Practice* (Englewood Cliffs, N.J. 1978) 56.

Profiling students before setting them loose in a computerized learning environment is one way of overcoming the opposition of freedom versus structure. Another way involves a different conception of the student's role. When we put students into a highly structured curriculum (whatever the technology used for packaging and delivery), we cast them as immature—not to say irresponsible—children. When we put students into a situation of freedom, we cast them as mature budding scholars, able to make their way responsibly through the material, different from the true master of the subject only in degree, not in kind.

A third role is also possible, at least in many humanistic disciplines—that of explorer. Explorers, whatever their age, are people who have invested their limited resources of time and money into travel to a self-selected place that they wish to get to know better. To this end, they typically put themselves into learning situations that are sometimes structured and sometimes unstructured. For example, before embarking, the typical explorer will consult with others who have already made the trip and will read books on a variety of topics and levels. That is, the explorer will consider a variety of structured approaches to mastering the subject and may well put together a personal synthesis of them all. Once arrived at the destination, the explorer continues to seek out structured learning—for example, a guided tour—while also continually breaking out of this routine by freely exploring the environment, making connections and discoveries along the way.

Applying this to multimedia computer programming, I would conclude by claiming that "links or stories" is an unnecessary choice—one we are not, in fact, forced to make. If the user is viewed as an explorer—not as a "student"—then at every moment, both guided tours ("stories") and free exploration ("making links") should be options provided by the program. The explorer metaphor is, of course, especially appropriate for a project such as the one I am working on (Cicero), where the student is, quite literally, an explorer sent back in time to a re-creation of ancient Rome. However, *mutatis mutandis* (as the Romans said), there is no reason why the metaphor could not be applied in many other contexts, even where the world of the program is not an imaginary trip to an exotic time and place.

16

Getting a Boost from the Real World

Ted Kaehler
Apple Computer, Inc.

17

"Direct manipulation" of things on the screen is a powerful idea, because it draws on experience we have in the real world. Once a person "gets it" that they can drag icons around on the Macintosh screen, you don't have to explain the details. They already know what to expect. Moving a file from one folder to another is very much like moving groceries from a paper bag to the refrigerator. We unconsciously make up little algorithms to do these tasks, and when we see a direct analogy on the computer screen, we can transfer our knowledge. We stop thinking out every step and start doing the actions unconsciously.

Programming languages have gotten a similar boost from the real world, but at a more abstract level. An "object" is a package of data and procedures that belong together. Object-oriented programming encourages the programmer to think of each part of his or her system as a "thing." When the programmer has divided the system into the right set of parts—the set that corresponds best to how he or she thinks about it on a visceral level—writing code becomes much easier. When the objects are aptly named, they behave as one would expect, and the programmer can skip a level or two of abstraction and mental translation. A package of data and procedures named a "push-down-stack" is much easier to use when it acts like a single "object," behaves as a stack should, and can be created simply by saying "make me a new stack."

One of the problems with object-oriented languages is that objects are difficult to edit directly. Objects are exactly what you want for constructing a card catalog for a library. However, it is very laborious to edit a card from the programming language (the program to change one field is more complex than the card). The time-honored solution is to build a specialized editor, so that the user can directly manipulate the data on the card. Building a specialized object editor is difficult—in fact, often harder than writing the original program. Since objects are hard to manipulate, they are scarce. The programmer is in the awkward position of wanting to play in a universe but having to create every blade of grass.

A hypertext system combined with an object-oriented language is a fusion of these worlds. All around you is a wealth of objects—paragraphs, links, documents, pictures, and spreadsheets. These objects are there because they are actually being used. At last there is something real to talk about, and with an object-oriented language, the talking is the best we know how to do. A little fragment of code is all one needs to say, "When I click on this button, check that this field contains a date and that this field is a legal call number for a book." Many objects, such as paragraphs, records, and pictures are already there in full bloom. The hypertext system provides a good editor "for free."

Laura Gould and Bill Finzer's Programming by Rehearsal and Dan Ingalls's Ariel are two early systems that allowed programmers to directly manipulate an object as soon as it was created (without building a specialized editor). Systems that allow the object-oriented ease of description and programming, but that also have the richness of data and the highly honed editors of a hypertext system will be quite compelling. I hope this fusion will help us to complete the unfulfilled vision—what the average person thought of when he or she first heard the words "personal computer."

Ted Kaehler, a Senior Engineer at Apple Computer, was educated at Stanford and Carnegie-Mellon (M.S. in Computer Science). He was a member of the SmallTalk group at Xerox P.A.R.C. for 14 years before joining Apple in 1985. Currently he is a member of the Advanced Technology Group at Apple.

Authoring with Hypermedia

Carol Kaehler
Apple Computer, Inc.

20

With hypermedia, authors have the ability to create large and complex learning environments that can be easily navigated and annotated by users. Two challenges in designing an interface for such an environment are as follows:

1. Showing users of the system where they are in relationship to the rest of the system—answering the inevitable "Where am I? Show me a map."

2. Preventing readers from being overwhelmed with the amount of information to be navigated.

Hypermedia at once causes and offers solutions for both problems. Because of the branching nature of hypermedia systems, it's easy to go down a path and lose sight of where you came from. You might be able to look back one level, or retrace single steps, or even see a map of the system. But a better solution is to design a system that includes an ever-present map that changes as you traverse it. For example, in a system that includes both text and graphics, you might use familiar graphic symbols such as notebook tabs that change as you move from subject to subject, creating the illusion that you're in a geographical location of a book—the reference chapter or the index, for example.

The second problem—making a large amount of information seem negotiable, is a natural outcome of using such built-in maps. The user doesn't see 1,000 forbidding pages of text, but rather a simple graphic notebook with seven or eight subjects. The structure belies the complexity within.

Hypermedia also brings serendipitous rewards. Entering the realm of hypermedia, authors might worry that readers will feel uneasy not knowing exactly how much of the book remains to be read. Readers are used to judging what they have yet to conquer in a subject by how many pages are behind them and how many are still ahead. But instead of this anticipated problem, hypermedia authors discover that you can hide bulk behind a simple interface. And so readers who might be overwhelmed by a thick book (or 800 kilobytes of information) see only a small, easily navigated notebook.

Carol Kaehler is a Senior Technical Writer at Apple Computer. She was educated as a musician (M.A. in Music Theory and Composition) but joined Apple in 1982 to write *Macintosh*, the owner's guide for the Macintosh computer. She also wrote the manuals for MacPaint and MacProject. Her most recent work at Apple has been in developing a hypertext help facility.

Navigation in Large Knowledge Spaces

Steven Weyer
Apple Computer, Inc.

22

Effective multimedia presentations are possible, but more a product of art than science. How can authoring environments provide the right amount of guidance and constraints so that creativity can be encouraged while special effects for their own sake are discouraged? As presentations become less canned and more interactive, personalizable and unanticipated, intelligence must be present to help in not only selecting (or generating) where to go next based on content, but also in making the transition between media styles of source and destination (e.g., the analog of film's cut, fade, or dissolve).

Navigating and learning in large knowledge spaces are central to both Vivarium (Alan Kay's project at Apple) and CYC (Doug Lenat's project at MCC). The Vivarium approach to information exploration is to create animal assistants—creatures with behavior, delegated plans, limited intelligence, and perhaps personalities—who act as surrogates in exploring a "knowledge grotto." CYC employs a more direct approach for the user by amplifying and creating sensory cues for the many dimensions of the knowledge space, and by suggesting filters to apply or axes to follow. Both are long-term, high-risk projects; however, such approaches are necessary if we are to deal with large knowledge bases in which we wish to deal with content as concepts rather than just literal text.